第一图书网, tushu007.com

# <<Flex 3权威指南>>

## 图书基本信息

书名：<<Flex 3权威指南>>

13位ISBN编号：9787115196903

10位ISBN编号：7115196907

出版时间：2009-3-1

出版时间：人民邮电出版社

作者：（美）塔伯（Tapper，J.）

页数：502

译者：杨博,杜昱宏

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：http://www.tushu007.com

第一图书网, tushu007.com

# <<Flex 3权威指南>>

## 前言

Adobe Flex作为一种跨操作系统和浏览器的开源框架，已经成为目前开发RIA最炙手可热的技术！本书是Adobe公司官方培训教材，通过一个完整的贴近实战的网上杂货店应用，用课程的形式，使读者身临其境地在40个小时内轻松掌握Flex开发涉及的各种技术和技巧。
学会构建用户体验绝佳的专业级应用程序。
书中详细介绍了Flex BIJilder、AIR、MXML-ActiollScript的概念，用户与后台数据交互的各种方式以及这些方式之间的区别，同时还讨论了如何使用Flex组件，如何编写你自己的Flex组件。

第一图书网, tushu007.com

# <<Flex 3权威指南>>

## 内容概要

本书是Adobe公司官方教程，以课程的形式详细介绍了如何用Flex 3开发一个完整的界面华丽的网上杂货店应用程序，其中包括一个用于查看销售细节实时图的管理图表，以及一个用于添加或编辑网上商店中要销售商品的数据输入程序。
书中详细介绍了Flex 3的相关知识点，具体包括构建、部署和调试RIA，使用功能强大的MXML语言布置应用程序，使用MXML和ActionScript构建自定义的UI组件，使用XML、基于SOAP的Web服务和远程调用协议AMF连接到数据等。

本书适合各层次Flash和Flex开发人员阅读。

第一图书网, tushu007.com


## 作者简介

JeffcTapper,Adobe技术社区最有影响力的专家之一,拥有丰富的Web开发经验,客户包括摩根士丹利、IBM、道琼斯等公司。
他还是WebcDevelopers&DesignerscJournal杂志的编委。

MichaelcLabriola,Adobe技术社区最著名的专家之一,是Adobe开发人员中心的重要撰稿人,也是Adobe技术邮件列表中最受欢迎的“大侠”。
除本书外,他还与JeffcTapper合著了《AIR权威指南》一书(人民邮电出版社即将出版)。

MatthewcBoles,Adobe公司客户培训团队的技术主管,他还参与设计了Adobe开发领域的官方课程。

JamescTalbot,Adobe公司资深工程师,有丰富的Flex开发和培训经验,帮助全世界许多用户使用Adobe技术创建了各种重要应用。
他也是ObjectcOrientedcActionScriptc2.0一书的作者。

第一图书网, tushu007.com

# <<Flex 3权威指南>>

## 书籍目录

第1课　理解RIA1.1　理解计算机应用程序的演化过程1.2　超越基于页面的架构1.3　明确RIA的优势1.3.1　企业经理1.3.2　IT组织1.3.3　终端用户1.4　各种RIA技术1.4.1　Ajax1.4.2　Flash1.4.3　Flex1.4.4　WPF、XAML、Silverlight和Expression1.5　学习收获第2课　开始2.1　开始Flex应用程序开发2.2　创建项目和MXML应用程序2.3　理解Flex Builder 3工作平台2.4　运行应用程序2.5　创建第二个应用程序页面，然后进入设计模式2.6　准备下一课2.7　学习收获第3课　布置界面3.1　学习容器相关知识3.2　用设计模式布置电子商务应用程序3.3　使用基于约束条件的布局3.3.1　使用相对父容器的约束条件3.3.2　使用增强型约束条件3.4　使用视图状态3.5　在源码模式中布置应用程序3.6　学习收获第4课　使用简单的控件4.1　简单控件简介4.2　显示图片4.3　建立详细信息视图4.4　使用数据绑定将数据结构链接到简单的控件4.5　使用Form布局容器布置简单的控件4.6　将单选按钮和数据字段添加到Dashboard4.7　学习收获第5课　处理事件和数据结构5.1　理解事件处理5.1.1　理解一个简单的例子5.1.2　用ActionScript函数处理事件5.1.3　调用事件处理函数时传递数据5.2　当creationComplete事件发生时建立数据结构5.3　使用来自事件对象的数据5.4　建立一个自定义的ActionScript类5.5　建立一个方法来创建对象5.6　建立购物车类5.7　学习收获第6课　为控件使用远程XML数据6.1　从HTTPService中获取XML数据6.1.1　创建HTTPService对象6.1.2　调用send()方法6.1.3　使用返回的数据6.1.4　理解安全问题6.2　通过HTTPService获取XML数据6.3　将HTTPService数据填入ArrayCollection6.3.1　使用ArrayCollection6.3.2　使用Collection作为数据提供者6.4　将数据填入ComboBox控件并用编程方法添加选项6.5　在Tree控件中使用XML数据6.5.1　理解E4X运算符6.5.2　将XML数据填入Tree控件6.6　获取XML数据并将这些数据转换为自定义类型的ArrayCollection6.7　对复杂数据结构使用数据绑定6.8　操作购物车数据6.8.1　将项目添加到购物车6.8.2　对ArrayCollection中的项目进行排序6.8.3　添加项目或者更新数量6.8.4　使用光标定位ShoppingCartItem6.9　添加Remove按钮6.10　学习收获第7课　用MXML创建组件7.1　介绍MXML组件7.1.1　理解创建自定义组件的基础知识7.1.2　一步步创建自定义组件7.1.3　在应用程序架构中使用自定义组件7.2　创建可更新/删除商品的组件，并将它实例化7.3　单击Update和Delete按钮时弹出商品信息7.4　创建另一个值对象7.5　为三个应用程序都创建数据管理器组件7.6　使用新的数据管理器组件7.7　实现添加商品的功能7.8　为Dashboard应用程序创建并使用组件7.9　学习收获第8课　使用控件和Repeater控制数据集8.1　使用数据集8.1.1　理解HorizontalList和TileList组件8.1.2　实现labelFuntcion8.1.3　实现itemRenderer8.2　使用HorizontalList和itemRenderer显示类别信息8.3　基于选中的类别显示商品8.3.1　使用Repeater遍历数据集8.3.2　从重复的组件中获取数据8.3.3　定位Repeater建立的组件8.3.4　理解TileList和Repeater的性能差异8.3.5　基于类别显示商品8.4　编写状态以显示商品信息详情8.5　将商品放入购物车8.6　学习收获第9课　使用自定义事件9.1　理解松耦合架构的优势9.2　分发事件9.3　为组件声明事件9.4　明确对自定义事件类的需求9.5　建立并使用CategoryEvent9.6　创建并使用ProductEvent类9.6.1　使用ProductEvent从购物车中移除商品9.6.2　使用ProductEvent向购物车添加商品9.7　理解事件流和事件冒泡9.8　学习收获第10课　用ActionScript 3.0创建自定义组件10.1　用ActionScript构建组件10.2　创建类的基本结构10.3　重写createChildren()方法10.3.1　在ActionScript中创建Button10.3.2　使用addChild()方法向组件添加Button10.3.3　理解边饰和rawChildren10.3.4　使用rawChildren上的addChild()向边饰部分添加元素10.4　在Flex中定位和调整组件大小10.5　理解measure()方法10.6　重写updateDisplayList()方法10.7　学习收获第11课　使用DataGrid与项目呈现器11.1　介绍DataGrid和项目呈现器11.2　向ChartPod添加通用的DataGrid11.3　向Dashboard中添加HTTPService调用11.4　用DataGrid显示购物车11.4.1　为DataGridColumn添加内联的编辑控件11.4.2　创建MXML的项目呈现器以显示商品11.4.3　创建内联的MXML项目呈现器用于显示Remove按钮11.4.4　使用set和get函数更新ShoppingCartItem11.5　使用AdvancedDataGrid11.5.1　为AdvancedDataGrid排序11.5.2　以专家模式进行排序11.5.3　设置AdvancedDataGrid的样式11.5.4　对数据进行分组11.5.5　显示摘要数据11.6　学习收获第12课　使用拖放12.1　介绍拖放操作管理器12.2　在两个DataGrid之间拖放12.3　在DataGrid和List之间拖放12.4　用不可以直接拖曳组件上实现拖放操作12.5　将商品拖进购物车12.6　学习收获第13课　实现导航13.1　导航13.2　在DataEntry程序中使用TabNavigator13.3　在电子商务应用程序中添加主页和结账页面13.4

第一图书网, tushu007.com


创建由ViewStack显示的结账过程的第一步13.5　使用ViewStack完成结账过程13.6　学习收获第14课　使用格式器和验证器14.1　介绍格式器和验证器14.2　用格式器类显示电子商务应用程序的货币信息14.3　使用验证器类14.4　用正则表达式来验证数据(第一部分)14.5　用正则表达式来验证数据(第二部分)14.6　构建自定义验证器类14.7　学习收获第15课　实现历史管理和深度链接15.1　历史管理简介15.2　实现Navigator容器的历史管理15.3　构建自定义的历史管理器15.4　介绍深度链接15.4.1　实现Flex 3的深度链接15.4.2　为程序添加深度链接15.5　学习收获第16课　自定义Flex应用程序的外观16.1　用样式和皮肤来设计16.2　使用样式16.2.1　用标签的属性设置样式16.2.2　理解样式继承16.2.3　用mx：Style标签设置样式16.2.4　用Flex Builder工具来编写CSS16.2.5　用CSS文件来设置样式16.3　在运行时修改CSS16.3.1　运行时修改CSS的优点16.3.2　从CSS文件创建SWF16.3.3　用StyleManager加载CSS的SWF16.3.4　使用加载的CSS覆盖样式16.4　皮肤组件16.4.1　图形皮肤16.4.2　导入CS3创建的皮肤16.4.3　可编程的皮肤16.5　学习收获第17课　使用Web服务17.1 服务器通信介绍17.2 使用远程服务器调用的事件模型17.3 配置应用程序在本地运行17.4 在Dashboard中使用Web服务17.4.1 处理Web服务的结果17.4.2 调用Web服务方法17.5 在DataEntry程序里使用Web服务17.6 使用Web Service Introspection向导17.7 在应用程序中使用生成代码17.8 用Flex Builder重构17.9 完成生成代码的集成17.10 更新和删除商品17.11 学习收获第18课　访问服务器端对象18.1 向服务器上传文件18.2 用RemoteObject来保存订单18.2.1 更新Flex编译器命令18.2.2 在结账过程分发订单确认事件18.2.3 创建和调用远程对象18.2.4 把ShoppingCart传入Checkout组件18.2.5 改变程序状态以将其返回Welcome页面18.3 把ActionScript对象映射到服务器对象18.4 Flex Builder数据向导18.5　学习收获第19课　可视化数据19.1 探索Flex图表组件19.1.1 图表类型19.1.2 图表软件包19.1.3 图表的组成部分19.2 布置最初的图表19.3 填充图表19.3.1 指定图表的系列19.3.2 向折线图和柱状图添加横轴和纵轴19.4 向图表添加图例19.5 限制轴上显示的标识数量19.6 与图表交互19.6.1 鼠标经过事件19.6.2 单击事件19.6.3 选中事件19.7 添加图表事件19.8 向图表添加动画19.9 用样式自定义图表外观19.10　学习收获第20课　创建模块化的程序20.1 介绍Flex 3的模块化程序20.2 使用Flex模块20.2.1 使用Module类20.2.2 用ModuleLoader标签来代替Checkout代码20.3 理解RSL20.4 使用Flex框架RSL缓存20.4.1 理解框架缓存的目的20.4.2 使用有签名的RSL与没有签名的RSL20.4.3 检查程序的当前大小20.4.4 配置程序使用框架RSL20.4.5 检查框架RSL带来的差别20.5 创建库项目20.5.1 把类和资源加入库20.5.2 在FlexGrocer程序中使用FlexGrocerLibrary20.6　学习收获第21课　部署Flex应用程序21.1 编译应用程序的部署版本21.2 用AIR摆脱浏览器21.2.1 开始AIR21.2.2 安装AIR21.2.3 安装第一个程序21.3 创建AIR应用程序21.3.1 创建新的AIR项目21.3.2 使Dashboard程序成为AIR应用程序21.3.3 用Application.xml文件定制程序21.3.4 导出AIR文件21.4　学习收获第22课　创建过渡和行为22.1 介绍行为和过渡22.1.1 在组件上使用行为22.1.2 在视图状态上使用过渡22.2 在组件上实现效果22.3 为视图状态添加效果22.4　学习收获第23课　Flex的打印功能23.1 介绍Flex的打印功能23.2 第一次使用Flex的打印功能23.3 在不可见容器中使用PrintDataGrid23.4 在单独的组件中构建打印视图23.5 调整打印输出的缩放比例23.6 从结账过程打印收据23.7　学习收获第24课　使用共享对象24.1 介绍共享对象24.1.1 创建共享对象24.1.2 读取共享对象24.2 构建SharedObject来保存购物车数据24.3 从已有的共享对象中读取数据24.4　学习收获第25课　调试Flex应用程序25.1 调试技术简介25.2 观察客户端与服务器端的数据交换25.3 学习更多调试器相关知识25.3.1 学习设置断点的更多相关细节25.3.2 在调试器中检查变量及其相关值25.4 使用try-catch处理错误25.4.1 使用try-catch语法25.4.2 理解错误类型25.4.3 使用多个catch块25.4.4 使用一个catch块的示例25.4.5 在第一个catch块中使用基类Error的不良做法示例25.4.6 使用finally语句25.4.7 使用throw语句25.4.8 创建自己的错误类25.5　学习收获第26课　对Flex应用程序进行剖析26.1 Flash Player的内存使用26.1.1 Flash Player的内存分配26.1.2 按引用传递和按值传递26.1.3 Flash Player的垃圾收集26.1.4 垃圾收集26.2 对Flex应用程序进行内存剖析26.2.1 审查ProfilerTest应用程序26.2.2 剖析ProfilerTest应用程序26.2.3 修复ImageDisplay类26.3 对Flex应用程序进行性能剖析26.3.1 剖析ProfilerTest应用程序26.3.2 修复ProfilerTest类26.4　学习收获附录A 安装指南

第一图书网, tushu007.com


## 章节摘录

第1课　理解RIA　　1.1　理解计算机程序的演化过程　　最早的商业应用程序的所有数据处理都发生在大型主机上，而客户端的任务仅仅包括显示来自服务器的信息以及接受用户输入，这主要是因为数据处理的成本很高。
要在整个企业中遍布强大的客户端是无法承受的，因此只能将所有的数据处理集中起来，并由“哑终端”负责用户交互。
随着内存和处理的成本越来越便宜，哑终端被微机（也叫C）所取代。
微机的处理能力更强，可以单独运行更多的桌面应用程序，比如文字处理程序和电子表格程序，因此，服务器不再是必需的了。
应用微机的组织所面临的一个挑战在于不容易集中数据。
虽然大型主机时代集中了一切，但是到了微机时代却要把数据分布开，结果在集中业务规则和同步企业内部数据时碰上了许多难题。
为了解决这些问题，多家厂商发布了力求结合微机与大型主机二者优势的平台，从而导致C／S（Client／Server，客户机朋艮务器）架构的诞生。
这些平台把微机的功能和易用性提供给终端用户，同时允许从一个集中地点存储和访问商业逻辑和数据，为用户解决了当时的那些问题。
但C／S体系引发了一个新问题——软件分发。
如果客户端应用程序发生变化，IT部门就不得不手动对每一台单机上的软件进行重装或升级。
许多公司发现，他们需要用一个全职的IT人员来负责保持终端用户桌面电脑上的软件是当前最新的。
随着20世纪90年代因特网的爆炸式发展，出现了一种商业应用程序的新模式。
这种模式依赖一个作为瘦客户端的Web浏览器，其主要职责是呈现HTML（Hypertext Markup Language，超文本标记语言）并把请求发回到应用服务器，而应用服务器动态地生成页面并传给客户端。
这往往被称为“基于页面的架构”①，该模式成功地解决了C／S时期面临的软件分发问题。
每次终端用户需要应用程序时都会从服务器下载，所以只需升级单个中心服务器，就能自动分发给整个用户基。
这一模式已经成功并将继续用于许多应用程序，但同时也带来了一些显著的缺点和局限性。
实际上，因特网应用程序与大型机应用程序有很多相似之处，比如，所有处理都集中在服务器，而客户端只是呈现数据和捕获用户反馈。
该模式最大的问题与UI（UserInterface，用户界面）有关。
过去10年内终端用户逐渐接受的很多便捷元素都丢失了，而ul也受到HTML能力的限制。
比如，桌面软件以及C／S应用程序会常常使用拖放功能。
然而，HTML应用程序几乎从不使用该功能，这是因为该功能太复杂，而且缺乏支持跨浏览器fl0DHTML（Dynamic HTML,动态HTML）元素，而DHTML元素却是在纯HTML／DHTML方案中实现拖放功能所必需的。
在大多数情况下，能建成并交付使用的解决方案的总复杂度是大大降低了。
虽然Web使部署应用程序更便捷，但基于Web的商业应用程序的功能却大幅度地后退了，这是因为基于浏览器的应用程序必须适应HTML以及HTTP（Hypertext Transport Protocol，超文本传输协议）等Web架构的局限性。
今天，对基于因特网应用程序的需求持续增长，与20世纪90年代中期的需求又有很大不同。
终端用户和企业进行因特网技术投资时会提出越来越高的要求。
为了给用户提供真正的价值，许多公司正在为因特网应用程序寻找更“丰富”的模式，既拥有传统桌面程序的丰富媒体，又拥有Web应用程序天生的丰富内容。
随着因特网应用程序开始被用于核心业务功能，应用程序的可维护性变得更加关键。
一个应用程序的可维护性直接关系到应用程序的架构。
可惜，许多Web应用程序在构建时几乎没有考虑过应用程序架构的原则，因此很难维护和扩展。
今天，可以使用许多方式来将业务、数据访问和显示三者分隔开，因而能更容易地为应用程序建立一

个稳固的架构。
随着web服务等元素的引入，SOA（Service—OrientedArchitecture，面向服务的架构）的概念对基于Web的应用程序来说已经逐渐弯得可行。
为了满足企业的需求，RIA必须能做到以下几点。
RIA必须能为代码执行、内容显示和网络通信提供一个高效率、高效能的运行时环境。
在1.2节，将会研究基于HTML的标准应用程序的各种限制；学习与基于页面的传统架构的大量性能相关的问题。
RIA必须能提供强大而又可扩展的对象模型以方便交互。
Web浏览器近年来的进步，已经有能力支持通过JavaScript和lDHTML在DOM（Document Object Model，文档对象模型）中进行交互，但它们仍然缺乏跨平台和跨浏览器的标准化支持。
若使用上述工具建立可以运行在各种浏览器和操作系统中的RIA，就要创建同一应用程序的多个版本。
RIA必须能使用服务器端对象，不论是通过web服务还是别的类似技术。
RIA要保证能够使显示逻辑和用户界面与封装在服务器端中的应用逻辑彻底地分离。
RIA必须允许离线使用因特网应用程序。
一方面，笔记本电脑和其他便携设备的人气持续增长，另一方面，因特网应用程序最严重的局限性之一，就是要求运行应用程序的机器必须连接到因特网。
虽然用户绝大多数时间都可以上网，但对商务旅行者来说，有些时候是不可能连接到因特网的。
一个成功的RIA应该允许用户不论在线或者离线都能使用它。
1.2 超越基于页面的架构 对于经验丰富的Web开发者来说，构建RIA最大的挑战之一是脱离基于页面的架构。
传统的Web应用都围绕着网页这个概念。
不论使用哪种服务器端技术（如果用到的话），流程总是类似下面这样。
（1）用户打开浏览器，向web服务器请求一个页面。
（2）wed服务器收到请求。
（3）（可选）网络服务器把请求交给应用服务器来动态组装页面。
（4）（可选）网络服务器从文件系统获取静态页面。
（5）web服务器将（动态或静态）页面发回给浏览器。
（6）浏览器绘制页面以替代先前显示的页面。
即使在新页面的大部分内容与先前页面基本相同的情况下，整个新页面依然需要被发送到浏览器并呈现。
这是传统Web应用程序低效的原因之一：每次用户交互都需要在浏览器中加载一个新页面。
RIA的关键目标之一就是要减少每次请求传送的额外数据量。
相比下载整个页面，为什么不可以只下载改变了的数据，然后再更新用户正在浏览的页面呢?这正是标准桌面程序或者C／S应用程序工作的方式。
虽然这个目标似乎不难，也很容易被首次进行RIA开发的开发者所接受，但Web开发者往往会把基于页面的思维方式带到RIA中，执著于想解决RIA中已经不需要解决的旧问题，例如如何“维持状态”。
举例来说，用户登录以后，怎么知道他们是谁，允许他们用应用程序做些什么? 维持状态是基于Web的应用程序引入的一个挑战。
HTTP被设计为一种无状态协议，每次对服务器的请求都是一次原子操作，对先前的请求一无所知。
这种无状态的本质让Web更具效率和冗余性，因为在浏览器和服务器之间并不需要保持连接。
每一个新页面的请求持续的时间仅仅是服务器用来检索和发送该页面的时间，这样能让单台服务器处理更多同时发来的请求。
Web无状态的本质也给应用程序开发者带来了一些挑战。
通常，应用程序需要记住用户相关的资料，如登录权限、添加到购物车的物品，等等。
如果没有能力记住一次次请求之中的这些数据，就不可能进行真正的应用开发。

第一图书网, tushu007.com


为了帮助解决这个问题，实施了一系列的解决方案，其基本原理是在每次请求时将一个独特的标识发回给服务器（通常使用cookie。
cookie是一些小文本文件，包含了应用程序指定的每个用户的特定标识符），并在服务器存储该用户信息。

有别于传统的Web应用程序，RIA可以绕过许多上述问题。
因为应用程序在被用户使用（而不像基于页面模式那样只是加载和卸载）的整个时间内一直待在客户端的内存中，所以只要设定一次变量，就能在整个应用程序的生命周期中访问该变量。

许多情况下，开发RIA时与开发Web应用程序时所需的思维方式稍有区别，处理状态的方法不同只是其中之一。
事实上，基于Web的RIA相LLWeb应用程序反而与C／S应用程序有更多的相似性。

1.3　明确RIA的优势　　与20世纪90年代中后期的网络泡沫时不同，企业不会再简单地因为因特网技术很酷就投资。
要取得成功，一项新技术必须体现出真正的投资回报率和真正的价值增长。
RIA可在以下几个层次上实现以上目标，其中都降低了开发成本并增加了整个组织的价值。

第一图书网, tushu007.com


## 媒体关注与评论

Adobe技术社区最有影响力的专家之一，拥有丰富的Web开发经验，客户包括摩根士丹利、IBM、道琼斯等公司。
他还是Web Developers&Designe．Jourma杂志的编委。
——Jeff Tapper　　Adobe技术社区最著名的专家之一，是Adobe开发人员中心的重要撰稿人，也是Adobe技术邮件列表中最受欢迎的“大侠”。
除本书外，他还与Jeff Tapper合著了《AlR权威指南》一书(人民邮电出版社即将出版)。
——Michael Labriola　　Adobe公司客户培训团队的技术主管，他还参与设计了Adobe开发领域的官方课程。
James Talbot Adobe公司资深工程师，有丰富的Flex开发和培训经验，帮助全世界许多用户使用Adobe技术创建了各种重要应用。
他也是Object Oriented Action Scnript 2．0一书的作者。
——Matthew BoIes

第一图书网, tushu007.com

# <<Flex 3权威指南>>

## 编辑推荐

《Flex 3权威指南》适合各层次Flash和Flex开发人员阅读。

AdobeFlex作为一种跨操作系统和浏览器的开源框架，已经成为目前开发RIA最炙手可热的技术！《Flex 3权威指南》是Adobe公司官方培训教材，通过一个完整的贴近实战的网上杂货店应用，用课程的形式，使读者身临其境地在40个小时内轻松掌握Flex开发涉及的各种技术和技巧。

学会构建用户体验绝佳的专业级应用程序。

书中详细介绍了FlexBIJilder、AIR、MXML和ActiollScript的概念，用户与后台数据交互的各种方式以及这些方式之间的区别，同时还讨论了如何使用Flex组件，如何编写你自己的Flex组件，以及如何编写高性能的代码等内容。

全面涵盖了Flex3的各种新特性。

第一图书网, tushu007.com



## 版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:http://www.tushu007.com